



①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

⑫ Off nl ungungsschrift
⑩ DE 199 45 900 A 1

⑤1 Int. Cl.⁷:
G 06 F 11/00
G 01 R 31/28

②1 Aktenzeichen: 199 45 900.2
②2 Anmeldetag: 24. 9. 1999
④3 Offenlegungstag: 19. 4. 2001

DE 199 45 900 A 1

⑦1 Anmelder:
Siemens AG, 80333 München, DE

⑦2 Erfinder:
Gläser, Winfried, Dipl.-Ing., 85570 Markt Schwaben, DE;
Amandi, Dirk, Dipl.-Phys., 81477 München, DE;
Angele, Manfred, Dipl.-Ing., 81379 München, DE;
Mircescu, Alexander, Dr.-Ing., 81379 München, DE

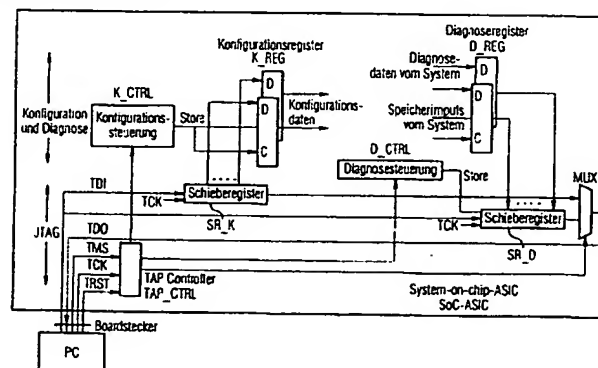
⑤6 Entgegenhaltungen:
DESPOSITO Joseph, New Debug Tools Give
Designers
Alternatives To In-Circuit Emulators, in:
ELECTRONIC DESIGN, 8.8.1999, S. 47,48,50,52,54;

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤4 Konfigurierung einer Trace-Schnittstelle über eine JTAG-Schnittstelle

⑤7 Bei einem SoC-ASIC (System on Chip-Application Specific Integrated Circuit) ist die Schnittstelle zur Nachverfolgung von Daten (Trace-Interface) über eine standardisierte JTAG-Schnittstelle konfigurierbar, wobei vorteilhafterweise Rückwirkungen auf das Echtzeitverhalten vermieden sind.



DE 199 45 900 A 1

Beschreibung

Der Anmeldungsgegenstand betrifft eine Anordnung, ein Verfahren und die Verwendung eines standardisierten JTAG-Interface's zur rückwirkungsfreien Nachverfolgung von Daten in einer System-on-Chip Anordnung.

Bei System on Chip (SoC) Architekturen sind verschiedene Komponenten, z. B. Mikroprozessoren, RAMs und komplexe HW Control Logik auf einem Chip angeordnet. Um ein SoC Design zu testen, ist es unabdingbar, in den Chip "hineinschauen" zu können, d. h. interne Datenströme aufzuzeichnen. Die Auswahl der aufzuzeichnenden Daten (Daten/Address Busse, Control Signale), sowie das Einstellen von Filtern und Triggerpunkten zur Datenreduktion ist notwendig, um die Trace Daten an die Testanforderungen anzupassen. Die Konfiguration des Tracers darf das Echtzeitverhalten des Systems in keiner Weise beeinflussen, da in bestimmten Anwendungen die Systeme redundant eingesetzt werden und somit ein Clock Cycle akurates Verhalten erfordern. Weiterhin verbieten die Stabilitätsanforderungen des Kunden jeden Eingriff in das Echtzeitverhalten.

Es ist notwendig, Diagnosedaten aus Registern nach außen zu geben und als Rückmeldungen an den Tracer zu geben (Regressives Testen). Dabei darf das Echtzeitverhalten des Systems in keiner Weise beeinflusst werden (rückwirkungsfrei). Diagnosedaten sind u. a. Statistik- und Fehlerinformationen.

Bisher war die Konfiguration/Einstellung des Tracers gar nicht möglich oder nur direkt nach System Reset und unter Beeinflussung des Echtzeitverhaltens. Das Auslesen der Diagnosedaten war bisher nur möglich, in dem das Echtzeitverhalten gestört wurde.

Dem Anmeldungsgegenstand liegt das Problem zugrunde, ein Verfahren zur Nachverfolgung von Daten in einem SoC-ASIC anzugeben, das die genannten Anforderungen unter Vermeidung der angegebenen Nachteile erfüllt.

Das Problem wird durch die in den Ansprüchen 1, 2 und 3 definierten Gegenstände gelöst.

Der Anmeldungsgegenstand nutzt aus, daß mit dem JTAG-Standardinterface Register geschrieben und gelesen werden können.

Beim Anmeldungsgegenstand erfolgt also die Konfiguration des Trace-Interface's asynchron/parallel durch das Beschreiben der Konfigurationsregister über ein standardisiertes JTAG Interface.

Da JTAG einen eigenen Takt TCK benutzt, ist das Schreiben und Lesen der Register vom Systemablauf unabhängig, das Echtzeitverhalten wird nicht gestört. Der anmeldungsgemäße Tracer ist nur ein Beobachter und damit passiv. Das Interface beim Anmeldungsgegenstand ist über eine Steuerungsschnittstelle, z. B. PC-Karte interaktiv bedienbar. Der Bediener kann im laufenden System beim Kunden die Verbindung zum JTAG Interface anstecken und interne Datenströme tracen und damit analysieren. Zusätzlich kann er im laufenden Betrieb die Statistik- und Fehlerinformationen auslesen und daraus Rückschlüsse auf weitere Konfigurationen des Tracers ziehen (regressives Testen).

Vorteilhafte Weiterbildungen des Anmeldungsgegenstandes sind in den Unteransprüchen angegeben.

Der Anmeldungsgegenstand wird im folgenden als Ausführungsbeispiel in einem zum Verständnis erforderlichen Umfang anhand von Figuren näher erläutert. Dabei zeigen:

Fig. 1 eine System-on-Chip-Anordnung mit einer Anordnung nach dem JTAG-Standard und

Fig. 2 einen Trace-Access-Punkt an einem Systembus.

In den Figuren bezeichnen gleiche Bezeichnungen gleiche Elemente.

Fig. 1 zeigt eine JTAG-Anordnung mit den Konfigurations-

ons- und Diagnoseregistern. Gesteuert wird JTAG von einem PC (Personal Computer) mit Einschubkarte.

Mit dem JTAG-Standardinterface können Register geschrieben und gelesen werden. Da JTAG einen eigenen Takt TCK benutzt, ist das Schreiben und Lesen der Register vom Systemablauf unabhängig, das Echtzeitverhalten wird nicht gestört.

Für die Konfigurierung des Tracers wird der Standard JTAG zum Beschreiben der Konfigurationsregister benutzt.

In den Konfigurationsregistern sind folgende Informationen speicherbar:

- a) Trace-Access-Point-Select (Auswahl des Daten-Aufzeichnungspunktes)
- b) Filterregister
- c) Triggernoderegister

Jeder Trace-Access-Point wird über ein Bit, dem Trace-Access-Point-Select, ausgewählt.

In den Filterregistern stehen Adreß-, Daten-, Control- und Statuswerte, die mit den aktuellen Werten an dem Beobachtungspunkt (Daten-Aufzeichnungspunkt) verglichen werden. Das Vergleichsergebnis wird vom Triggercontroller TR_CTRL ausgewertet. Die Werte sind:

Adreßwert: Adresse auf dem Adress-Bus

Datenwert: Datenwert auf dem Daten-Bus, Bitfeld in der ATM-Zelle oder Sensorwert (Kfz-Anwendung)

Controlwert: Buscontrol, z. B. Lesen, Schreiben, Burst

Statuswert: Zustand der Finite-State-Maschine, Fehlercodes

Im Filterregister kann jede Speicherstelle drei Werte annehmen: 0, 1 oder don't care, um Bereiche zu maskieren.

In den Triggernoderegistern wird gespeichert, wie das Vergleichsergebnis zu interpretieren ist, d. h. ob die Information des Trace-Access-Points weiter gegeben wird, wenn das Vergleichsergebnis >, < oder gleich ist. Zusätzlich wird gespeichert, ob das Resultat des Vergleichsergebnisses den Trace an dem Trace-Access-Point startet oder anhält.

Auch zum Auslesen langfristiger Daten aus den Diagnoseregistern wird eine JTAG-Konfiguration benutzt. Langfristige Daten sind Informationen der SoC-Komponenten wie Status-, Fehler- und Statistikinformationen. Sie werden vom System in Diagnoseregister geschrieben.

Die Diagnoseinformationen sind:

Statusinformation: Auslastung der Systembusse, Auslastung der Buffer usw.

Fehlerinformation: Fehler wie z. B. Synchronisationsfehler, Taktausfälle bei Interfaces, Overflow von Puffern

Statistikinformation: Häufigkeit von Fehlern.

Das JTAG Interface kann interaktiv über eine Steuerungsschnittstelle, z. B. PC(Personal Computer)-Karte bedient werden. Der Bediener kann im laufenden System beim Kunden die Verbindung zum JTAG Interface anstecken und interne Datenströme tracen und damit analysieren. Zusätzlich kann er im laufenden Betrieb die Statistik- und Fehlerinformationen auslesen.

Die Software auf dem PC(Personal Computer) ist so erweiterbar, daß der PC die Diagnoseregister liest und bei Bedarf selbstständig den Tracer so konfiguriert, daß er gezielt Daten ansieht, um die Diagnose zu verfestigen (Regressives Testen).

Die in Fig. 1 dargestellte JTAG-Anordnung wird, wie bereits erwähnt, von einem PC (Personal Computer) mit Einschubkarte gesteuert.

Die Daten-Aufzeichnungspunkt-Steuerung TAP_CTRL (für: Trace-Access-Point- Controller) erkennt den Befehl, daß die Konfigurationsregister K_REG geladen werden sollen. Sie gibt ein Signal an die Konfigurationssteuerung K_CTRL, die die Konfigurationsinformation aus dem

Schieberegister SR_K in die Konfigurationsregister K_REG schreibt.

In einer bevorzugten Ausführungsform werden die Konfigurationsregister alle gleichzeitig beschrieben. Das hat den Vorteil, daß keine aufwendige Kodierung notwendig ist. 5

Die Daten-Aufzeichnungspunkt-Steuerung TAP_CTRL erkennt den Befehl, daß der Inhalt der Diagnoseregister D_REG in das Schieberegister SR_D gespeichert werden soll. Sie gibt ein Signal an die Diagnosesteuerung D_CTRL, die den Inhalt der Diagnoseregister in das Schieberegister 10 SR_D schreibt.

Anwendungsbeispiele für die Konfigurierung

Ein SoC-ASIC (Application Specific Integrated Circuit) 15 weist im allgemeinen ein oder mehrere Prozessoren, Speicher, Empfangsteil und Sendeteil auf. Die Komponenten sind über Systembusse, Adress-Bus A_BUS, Datenbus D_BUS verbunden.

Fig. 2 zeigt ein Beispiel eines Trace-Access-Points TAP 20 an einem Systembus.

Der TAP bildet ein Trigger-Start-Signal TR_ST und ein Trigger-Stop-Signal TR_SP zur Herausfilterung einer gewünschten Information.

Immer wenn der Adreßbereich zwischen dem unteren und oberen Adreßbereich liegt, wird die Businformation (Adresse und Daten) aus dem SoC-ASIC herausgeschrieben. 25

In dem Triggermoderegister TM_REG ist der Betriebsmode gespeichert. Die beiden Adreßwerte sind in je einem 30 Konfigurationsregister K_REG1, K_REG2 gespeichert. Wenn die Busadresse in dem Adreßbereich liegt, gibt der Triggercontroller TR_CTRL das Signal Triggerstart TR_ST ab. Gleichzeitig wird die Businformation im Trace-Register gespeichert. Liegt die Adresse nicht in dem Adreßbereich, 35 wird das Signal Triggerstopp TR_SP aktiviert.

Ein weiteres Beispiel ist die Konfiguration des Tracers am Empfangsteil.

In einem Empfangsteil für die Signalisierung über ATM(Asynchronous Transfer Mode) zwischen Telekommunikations-Vermittlungsstellen kann z. B. der Filter die virtuelle Kanalnummer oder Zellinformation vergleichen und dann den Tracer triggern.

Bei einem SoC Design in der Kfz-Anwendung kann z. B. auf den Inhalt eines Sensorsignals gefiltert werden. 45

Der Filter am Sendeteil für die Signalisierung über ATM kann z. B. die virtuelle Kanalnummer oder Zellinformation vergleichen und auf Grund des Ergebnisses den Tracer triggern.

Bei einem SoC-Design in Embedded-Anwendung kann 50 z. B. auf Reglereinstellungen gefiltert werden.

Es ist Stand der Technik, daß Prozessoren ihren Programmcounter und Status (JTAG-Standard) rausgeben, um den Programmablauf zu verfolgen. Auch hier können Filter eingesetzt werden, die den Programmcounter vergleichen 55 und auf Grund des Vergleichsergebnisses den Tracer triggern.

Patentansprüche

60

1. Anordnung zur rückwirkungsfreien Nachverfolgung von Daten und deren Zuständen bei der

- eine auf einem Halbleitersubstrat realisierte Schaltung in einem Baustein angeordnet ist
- die Schaltung mit mehreren Komponenten gebildet ist, die über mindestens einen Systembus verbunden sind

die Schaltung eine mehrere Konfigurationsregi-

ster aufweisende Einrichtung zum Herausführen von Nachverfolgungs (Trace)-Informationen über eine Nachverfolgungsschnittstelle (Trace-Interface) aus dem Baustein aufweist

- die Konfigurationsregister über ein auf der integrierten Schaltung angeordnete, standardisierte JTAG-Schnittstelle schreibbar sind.

2. Verfahren zur rückwirkungsfreien Nachverfolgung von Daten und ihrer Zustände in einer System-On-Chip-Anordnung demzufolge die Schnittstelle zur Nachverfolgung der Daten (Trace-Interface) über ein standardisiertes JTAG-Interface konfigurierbar ist.

3. Verfahren nach Anspruch 2 dadurch gekennzeichnet, dass mehrere über die Schnittstelle zur Nachverfolgung der Daten (Trace-Interface) ansteuerbare Konfigurationsregister gleichzeitig konfiguriert werden.

4. Verwendung eines an sich bekannten JTAG-Interface's zur Konfigurierung einer Nachverfolgungsschnittstelle (Trace-Interface) in einer System-On-Chip-Anordnung.

Hierzu 2 Seite(n) Zeichnungen

- Leerseite -

THIS PAGE BLANK

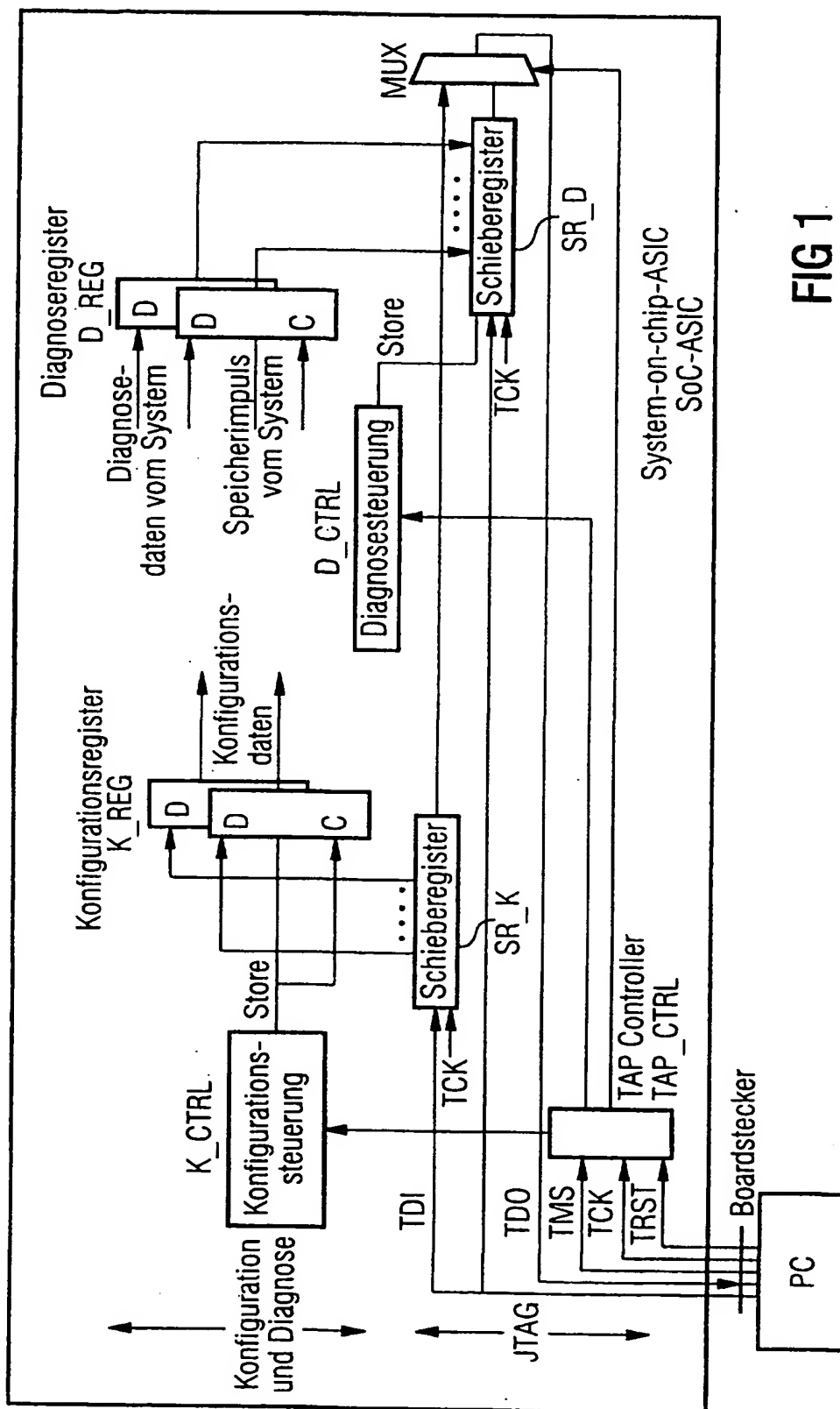


FIG 1

FIG 2

